

# ΣΥΣΤΗΜΑ ΔΙΑΧΕΙΡΙΣΗΣ ΙΑΤΡΙΚΟΥ ΦΑΚΕΛΟΥ ΣΕ ΝΟΣΟΚΟΜΕΙΟ



Λατσίνογλου Νικόλαος	42033
----------------------	-------

Ντόκος Στέφανος	42060
-----------------	-------

Τέμενος Νίκος	40975
---------------	-------

# Contents

<b>1.</b>	<b>ΕΙΣΑΓΩΓΗ .....</b>	<b>3</b>
1.1	ΣΚΟΠΟΣ ΤΟΥ ΚΕΙΜΕΝΟΥ .....	3
1.2	ΠΡΟΟΡΙΣΜΕΝΟ ΑΚΡΟΑΤΗΡΙΟ ΚΑΙ ΠΡΟΤΑΣΕΙΣ ΑΝΑΓΝΩΣΗΣ .....	3
1.3	ΣΤΟΧΟΣ ΚΑΙ ΕΥΡΟΣ ΤΟΥ ΠΡΟΪΟΝΤΟΣ .....	3
<b>2.</b>	<b>ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ .....</b>	<b>4</b>
2.1	ΠΡΟΕΛΕΥΣΗ ΚΑΙ ΠΡΟΟΠΤΙΚΗ ΤΟΥ ΠΡΟΪΟΝΤΟΣ .....	4
2.2	ΓΕΝΙΚΗ ΠΕΡΙΓΡΑΦΗ ΤΩΝ ΛΕΙΤΟΥΡΓΙΩΝ ΤΟΥ ΠΡΟΪΟΝΤΟΣ .....	4
2.3	ΚΑΤΗΓΟΡΙΕΣ ΧΡΗΣΤΩΝ .....	4
2.4	ΠΕΡΙΒΑΛΛΟΝ ΛΕΙΤΟΥΡΓΙΑΣ.....	5
2.5	ΠΕΡΙΟΡΙΣΜΟΙ ΣΤΗ ΣΧΕΔΙΑΣΗ ΚΑΙ ΤΗΝ ΥΛΟΠΟΙΗΣΗ.....	5
2.6	ΤΕΚΜΗΡΙΩΣΗ ΓΙΑ ΤΟ ΧΡΗΣΤΗ.....	5
2.7	ΠΡΟΫΠΟΘΕΣΕΙΣ – ΕΞΑΡΤΗΣΕΙΣ.....	5
<b>3.</b>	<b>ΑΠΑΙΤΗΣΕΙΣ ΓΙΑ ΤΙΣ ΕΞΩΤΕΡΙΚΕΣ ΔΙΕΠΑΦΕΣ .....</b>	<b>6</b>
3.1	ΔΙΕΠΑΦΕΣ ΧΡΗΣΤΗ.....	6
3.2	ΔΙΕΠΑΦΕΣ ΛΟΓΙΣΜΙΚΟΥ.....	7
<b>4.</b>	<b>ΛΕΙΤΟΥΡΓΙΕΣ ΣΥΣΤΗΜΑΤΟΣ.....</b>	<b>7</b>
4.1	ΚΑΤΑΧΩΡΗΣΗ ΝΕΟΥ ΑΣΘΕΝΟΥΣ.....	7
4.2	ΠΡΟΒΟΛΗ ΚΑΡΤΕΛΑΣ ΑΣΘΕΝΟΥΣ.....	8
4.3	ΚΑΤΑΧΩΡΗΣΗ ΕΡΓΑΣΤΗΡΙΑΚΩΝ ΕΞΕΤΑΣΕΩΝ .....	9
4.4	ΑΝΑΖΗΤΗΣΗ ΕΞΕΤΑΣΕΩΝ ΣΕ ΚΑΡΤΕΛΑ ΑΣΘΕΝΟΥΣ.....	10
4.5	ΑΛΛΑΓΗ (ΔΙΟΡΘΩΣΗ) ΣΤΟΙΧΕΙΩΝ ΑΣΘΕΝΟΥΣ.....	11
	Περιγραφή και προτεραιότητα.....	11
<b>5.</b>	<b>ΜΗ ΛΕΙΤΟΥΡΓΙΚΕΣ ΑΠΑΙΤΗΣΕΙΣ .....</b>	<b>12</b>
5.1	ΕΠΙΔΟΣΕΙΣ.....	12
5.2	ΑΣΦΑΛΕΙΑ ΠΛΗΡΟΦΟΡΙΑΣ .....	12
5.3	ΠΟΙΟΤΗΤΑ ΛΟΓΙΣΜΙΚΟΥ .....	12
<b>6.</b>	<b>ΔΙΑΓΡΑΜΜΑΤΑ UML .....</b>	<b>14</b>
6.1	ΔΙΑΓΡΑΜΜΑ ΚΛΑΣΕΩΝ (CLASS DIAGRAM) .....	14
6.2	ΔΙΑΓΡΑΜΜΑΤΑ ΑΚΟΛΟΥΘΙΑΣ (SEQUENCE DIAGRAM).....	15
6.3	ΔΙΑΓΡΑΜΜΑΤΑ ΔΡΑΣΤΗΡΙΟΤΗΤΑΣ.....	17
<b>7.</b>	<b>PMD TESTING .....</b>	<b>20</b>
<b>8.</b>	<b>UNIT TESTING &amp; CODE COVERAGE.....</b>	<b>21</b>
<b>9.</b>	<b>DEBUGGING .....</b>	<b>23</b>
<b>10.</b>	<b>ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΗΣ ΧΡΗΣΤΗ .....</b>	<b>13</b>
<b>11.</b>	<b>USER ACCEPTANCE TEST (UAT) PLAN.....</b>	<b>24</b>
11.1	DOCUMENT MANAGEMENT .....	24
11.1.1	Contributors .....	24

11.1.2	Version Control.....	25
11.2	USER ACCEPTANCE TESTING.....	25
11.2.1	User Acceptance Definition .....	25
11.2.2	UAT Responsibilities .....	25
11.3	UAT STRATEGY .....	26
11.4	USER TESTING.....	26

# **1. Εισαγωγή**

## **1.1 Σκοπός του κειμένου**

Ο κύριος σκοπός του παρόντος εγγράφου, είναι ο προσδιορισμός όλων των απαιτήσεων του λογισμικού διαχείρισης ιατρικού φακέλου σε νοσοκομείο, το οποίο είναι ένα πρόγραμμα που επιτρέπει σε υπαλλήλους νοσοκομείου να διαχειρίζονται τα ιατρικά αλλά και κυρίως τα προσωπικά στοιχεία από ασθενείς σε ένα νοσοκομείο. Το είδος των χρηστών συμπεριλαμβάνει όλους εκείνους που ασχολούνται με τους ιατρικούς φακέλους και έχουν άμεση ανάμειξη με αυτούς, δηλαδή τη γραμματεία του νοσοκομείου, τους νοσοκόμους και τέλος τους γιατρούς. Το πρόγραμμά μας σχεδιάστηκε έχοντας ως κύριο στόχο την εφαρμογή του σε νοσοκομεία, όμως θα μπορούσε κάλλιστα να χρησιμοποιηθεί και σε κλινικές για τον ίδιο σκοπό αφού ακολουθεί την ίδια φιλοσοφία λογισμικού.

## **1.2 Προορισμένο ακροατήριο και προτάσεις ανάγνωσης**

Αυτό το έγγραφο απευθύνεται στους χρήστες της εφαρμογής, δηλαδή τη γραμματεία, τους νοσηλευτές και τους γιατρούς. όμως παρέχει επιπλέον χρήσιμες πληροφορίες που μπορούν να αξιοποιηθούν μελλοντικά από προγραμματιστές και συντηρητές για την βελτίωση ή την περαιτέρω ανάπτυξη του. Επίσης είναι κρίσιμη η ανάγνωση από τον υποδιευθυντή του νοσοκομείου όπως επίσης και από τους εκπροσώπους του ιατρικού αλλά και του νοσηλευτικού προσωπικού έτσι ώστε να υπάρχει επικύρωση του λογισμικού από τους ενδιαφερόμενους αλλά και να μην υπάρχουν επιπλοκές σε περίπτωση που το πρόγραμμα αλλάξει μελλοντικά ή χρειαστούν τυχόν τροποποιήσεις με βάση τις ανάγκες του νοσοκομείου.

## **1.3 Στόχος και εύρος του προϊόντος**

Ο στόχος του προγράμματος για την δημιουργία ιατρικού φακέλου ασθενών είναι η εύκολη και απλή άντληση δεδομένων για τον ασθενή, η οποία γίνεται αυτόματα χωρίς να χρειάζεται περαιτέρω ενέργεια από το χρήστη πλην του πατήματος κουμπιών, αλλά και η λιτή εμφάνισή τους η οποία καθιστά το πρόγραμμα φιλικό προς το χρήστη.

## **1.4 Αναφορές**

1. Πρωτόκολλο τήρησης Ιατρικού Φακέλου
2. Ιατρικά πρωτόκολλα
3. Νοσηλευτικά πρωτόκολλα

## **2. Γενική Περιγραφή**

### **2.1 Προέλευση και προοπτική του προϊόντος**

Το πρόγραμμα διαχείρισης ιατρικού φακέλου σε νοσοκομείο δεν αποτελεί μια καινούργια δουλειά για εμάς μιας και έχουμε ήδη αναπτύξει αντίστοιχο λογισμικό για κλινικές αλλά και κάποια παρεμφερή για δημόσιους φορείς. Επίσης έχουμε δημιουργήσει παλαιότερα και άλλα λογισμικά διαφορετικού ύφους για τράπεζες και εστιατόρια.

### **2.2 Γενική περιγραφή των λειτουργιών του προϊόντος**

Οι λειτουργίες που θα εκτελεί το σύστημα μας είναι οι εξής:

- Δημιουργία ιατρικού φακέλου
- Προβολή ιατρικού φακέλου
- Αναζήτηση στο ιστορικό του ιατρικού φακέλου
- Εισαγωγή νέων στοιχείων σε ιατρικό φάκελο
- Προβολή/Αλλαγή των στοιχείων του ασθενούς σε ιατρικό φάκελο

### **2.3 Κατηγορίες χρηστών**

Στο προσωπικό που θα διαχειρίζεται το πρόγραμμα θα δωθούν κωδικοί πρόσβασης όμως ανάλογα με τη κατηγορία των χρηστών θα παρέχει περιορισμένη πρόσβαση στις λειτουργίες του προγράμματος. Αναλυτικότερα:

❖ Η γραμματεία θα έχει δικαίωμα να διαχειρίζεται τα εξής:

- Δημιουργία ιατρικού φακέλου
- Προβολή ιατρικού φακέλου
- Αναζήτηση στο ιστορικό του ιατρικού φακέλου
- Εισαγωγή νέων στοιχείων σε ιατρικό φάκελο
- Προβολή/αλλαγή των στοιχείων σε ιατρικό φάκελο

❖ Οι γιατροί θα έχουν δικαίωμα να διαχειρίζονται τα εξής:

- Προβολή ιατρικού φακέλου
- Αναζήτηση στο ιστορικό ιατρικού φακέλου
- Εισαγωγή νέων στοιχείων σε ιατρικό φάκελο

❖ Οι νοσηλευτές-τριες θα έχουν δικαίωμα να διαχειρίζονται τα εξής:

- Προβολή ιατρικού φακέλου.
- Αναζήτηση στο ιστορικό ιατρικού φακέλου.

## **2.4 Περιβάλλον λειτουργίας**

Το πρόγραμμά μας είναι σχεδιασμένο να δουλεύει πάνω σε λειτουργικό σύστημα Microsoft Windows 8.1 το οποίο είναι εγκατεστημένο στους υπολογιστές όλων των χρηστών του νοσοκομείου. Η βάση δεδομένων που χρησιμοποιεί το πρόγραμμα βρίσκεται σε Microsoft Windows Server με ταχύτητα επεξεργαστή στα 2.4GHz και μέγιστη διαθέσιμη μνήμη στα 8GB. Δεν χρειάζεται κάποια παραπάνω απαίτηση όσο αναφορά την υπολογιστική δύναμη των τερματικών αφού το πρόγραμμα σχεδιάστηκε για να λειτουργεί και σε υπολογιστές οι οποίοι δεν είναι νέας τεχνολογίας.

## **2.5 Περιορισμοί στη σχεδίαση και την υλοποίηση**

Το λογισμικό θα εκτελείται τοπικά σε κάθε υπολογιστή ο οποίος θα χρειαστεί να είναι εφοδιασμένος με λογισμικό Microsoft Windows 8.1 και όλοι οι υπολογιστές θα συνδέονται με μία βάση δεδομένων η οποία θα υπάρχει σε έναν Microsoft Windows Server με ταχύτητα επεξεργαστή στα 2.4 GHz και μέγιστη μνήμη στα 8 GB. Το πρόγραμμα θα αλληλεπιδρά με τα ήδη υπάρχον προγράμματα για εμφάνιση των αποτελεσμάτων για τις εργαστηριακές εξετάσεις και τις ιατρικές εικόνες (ακτινογραφίες). Όσο αναφορά τους περιορισμούς, οι υπολογιστές που βρίσκονται στη γραμματεία θα χρειαστεί να έχουν ταχύτητα 1 Gbps ενώ για τους γιατρούς και τους νοσοκόμους η ταχύτητα είναι στα 100 Mbps.

## **2.6 Τεκμηρίωση για το χρήστη**

- Μαζί με το λογισμικό θα δοθεί το κατάλληλο user manual ώστε να μπορεί να γίνει κατανοητή η λειτουργία του.
- Όταν θα παραδοθεί το πρόγραμμα στη τελική του μορφή θα γίνει από εμάς μια μικρή επίδειξη των δυνατοτήτων του προγράμματος για την κατανόηση της λειτουργίας του.

## **2.7 Προϋποθέσεις – Εξαρτήσεις**

Για την λειτουργία του προγράμματος κρίνεται απαραίτητο να υπάρχει σε κάθε παρεμπλεκόμενο ένα τερματικό όπως επίσης και απαιτείται η σύνδεση με το server έτσι ώστε να γίνονται οι καταχωρήσεις στη βάση δεδομένων αλλά και η πρόσβαση στο διαδίκτυο για να μπορέσουν να καταχωρούνται τα

δεδομένα στη βάση. Οποιαδήποτε από τις παραπάνω απαιτήσεις λείπει, παύει να είναι εφικτή η λειτουργία του προγράμματος.

## 3. Απαιτήσεις για τις εξωτερικές διεπαφές

### 3.1 Διεπαφές χρήστη

Η διεπαφή του χρήστη είναι ένα GUI που θα περιέχει τα εξής:

- ✓ Αρχικό παράθυρο login με όνομα και κωδικούς για κάθε χρήστη
- ✓ Παράθυρο με αποτελέσματα που εμφανίζει ανάλογα μηνύματα και κουμπιά για τις επιλογές αφού γίνει σωστή σύνδεση.  
Ανάλογα τον χρήστη εμφανίζονται τα εξής κουμπιά:
  - Γραμματεία 5 κουμπιά
    - Δημιουργία
    - Προβολή
    - Αναζήτηση στο ιστορικό
    - Εισαγωγή νέων στοιχείων
    - Προβολή/αλλαγή των στοιχείων
  - Γιατροί 3 κουμπιά
    - Προβολή
    - Αναζήτηση στο ιστορικό
    - Εισαγωγή νέων στοιχείων
  - Νοσηλευτές
    - Προβολή
    - Αναζήτηση στο ιστορικό
  - Κουμπί εκκαθάρισης της οθόνης (διαθέσιμο σε όλους τους χρήστες)
- ✓ Κάθε επιλογή κάνει ανακατεύθυνση σε καινούργιο παράθυρο έτσι ώστε να επιτευχθεί η λειτουργία του
- ✓ Όλα τα παράθυρα διαθέτουν κουμπί τερματισμού αλλά και ελαχιστοποίησης του προγράμματος

### 3.2 Διεπαφές λογισμικού

Το πρόγραμμα που υλοποιήθηκε συνδέεται με 2 ήδη υπάρχουσες εφαρμογές: α) ένα σύστημα διαχείρισης αποτελεσμάτων των εργαστηριακών εξετάσεων και β) ένα σύστημα διαχείρισης αποτελεσμάτων των απεικονιστικών μηχανημάτων (ιατρικές εικόνες). Τα αποτελέσματα των εφαρμογών αυτών θα εμφανίζονται στο παράθυρο του προγράμματος που υλοποιήθηκε. Επίσης θα πρέπει να υπάρχει σύνδεση με το server αλλά και σύνδεση με τη βάση δεδομένων αυτού έτσι ώστε να αντλούνται τα δεδομένα που έχουν αποθηκευτεί για κάθε ασθενή.

## 4. Λειτουργίες συστήματος

Το πρόγραμμά μας θα είναι προσβάσιμο από τους χρηστές με ένα απλό διπλό κλικ πάνω στο εκτελέσιμο αρχείο. Πατώντας το διπλό κλικ ο χρήστης έρχεται σε επαφή με τη πρώτη φόρμα του προγράμματος όπου γίνεται η διαδικασία του log in. Μετά την επιτυχή σύνδεση του χρήστη κλείνει το παράθυρο σύνδεσης αυτόματα και ανοίγει ένα νέο το οποίο περιέχει τις ακόλουθες λειτουργίες που μπορεί να εκτελέσει ο χρήστης. Είναι σημαντικό να αναφερθεί ότι ανάλογα με τη κατηγορία του χρήστη εμφανίζονται και οι κατάλληλες επιλογές. Αυτές είναι:

### 4.1 Καταχώρηση νέου ασθενούς

#### Περιγραφή και προτεραιότητα

Σε περίπτωση που θέλει ο χρήστης να κάνει καταχώρηση νέου ασθενούς θα χρειαστεί να πατήσει ένα κλικ στο κουμπί **create** όπου αυτόματα θα ανοίξει ένα νέο παράθυρο στο οποίο μπορεί να κάνει καταχώρηση του ασθενούς στο σύστημα με τα εξής στοιχεία:

- Όνομα
- Επίθετο
- Ημερομηνία Γέννησης
- ΑΦΜ
- Διεύθυνση
- Τηλέφωνο



### Ακολουθίες ερεθίσματος / απόκρισης

**Ερέθισμα:** Ο χρήστης θα πατήσει το κουμπί create για να δημιουργήσει ένα νέο ασθενή

**Απόκριση:** Το πρόγραμμα θα ανοίξει αυτόματα ένα νέο παράθυρο εισαγωγής στοιχείων νέου ασθενούς.

**Ερέθισμα:** Ο χρήστης συμπληρώνει τα στοιχεία του ασθενούς ώστε να δημιουργηθεί η νέα καρτέλα του εκάστοτε ασθενούς και πατάει το κουμπί προσθήκη.

**Απόκριση:** Το σύστημα με βάση το πεδίο ΑΦΜ κάνει έλεγχο για περίπτωση διπλοεγγραφής. Σε περίπτωση που δε βρει ότι υπάρχει ήδη καταχωρημένο το ΑΦΜ του ασθενούς δημιουργεί τη νέα καρτέλα. Ανεξαρτήτως αποτελέσματος εμφανίζεται αντίστοιχο μήνυμα.

## 4.2 Προβολή καρτέλας ασθενούς

### Περιγραφή και προτεραιότητα

Σε περίπτωση που θέλει ο χρήστης να δει μια υπάρχουσα καρτέλα κάποιου ασθενούς θα χρειαστεί να πατήσει ένα κλικ στο κουμπί **View** όπου αυτόματα θα δημιουργηθεί ένα νέο παράθυρο στο οποίο μπορεί να κάνει αναζήτηση του ασθενούς στο σύστημα με βάση το ΑΦΜ του

Προτεραιότητα: Υψηλή – 1.

### Ακολουθίες ερεθίσματος / απόκρισης

**Ερέθισμα:** Ο χρήστης θα πατήσει το κουμπί **View** για να αναζητήσει έναν ασθενή.

**Απόκριση:** Το πρόγραμμα θα ανοίξει αυτόματα ένα νέο παράθυρο όπου θα ζητά να δοθεί το ΑΦΜ του ασθενή.

**Ερέθισμα:** Ο χρήστης θα εισάγει το ΑΦΜ του ασθενή που ψάχνει και έπειτα το κουμπί προβολή.

**Απόκριση:** Το σύστημα με βάση τον ΑΦΜ θα κάνει την αναζήτηση του χρήστη στη βάση δεδομένων του προγράμματος και αν βρει αντιστοιχία στο ΑΦΜ θα προβάλει τα στοιχεία του. Σε διαφορετική περίπτωση εμφανίζεται μήνυμα αποτυχίας

### 4.3 Καταχώρηση εργαστηριακών εξετάσεων

#### Περιγραφή και προτεραιότητα

Σε περίπτωση που θέλει ο χρήστης να κάνει καταχώρηση εργαστηριακών εξετάσεων θα χρειαστεί να πατήσει ένα κλικ στο κουμπί **Update όπου** αυτόματα θα δημιουργηθεί ένα νέο παράθυρο στο οποίο μπορεί να κάνει καταχώρηση των εξετάσεων με τα εξής στοιχεία:

- Ούρα
- Αιματολογικές Εξετάσεις
- Χρονολογία Εξετάσεων
- ΑΦΜ

Προτεραιότητα: Υψηλή – 1

#### Ακολουθίες ερεθίσματος / απόκρισης

**Ερέθισμα:** Ο χρήστης θα πατήσει το κουμπί **Update** για καταχώρηση εργαστηριακών εξετάσεων.

**Απόκριση:** Το πρόγραμμα θα ανοίξει αυτόματα ένα νέο παράθυρο όπου θα ζητά να πατήσει ο χρήστης **γ**(yes) ή **η**(no) στα πεδία ούρα και αιματολογικές εξετάσεις. Επίσης χρειάζεται και η χρονολογία που έγιναν οι εξετάσεις για τη σωστή ταξινόμηση τους και τέλος το ΑΦΜ όπου με βάσει αυτό οι εξετάσεις θα προστεθούν στην κατάλληλη καρτέλα του ασθενούς που αντιστοιχεί το ΑΦΜ.

**Ερέθισμα:** Ο χρήστης θα εισάγει τα στοιχεία που χρειάζεται να πατήσει το κουμπί προσθήκη.

**Απόκριση:** Το πρόγραμμα με βάση το ΑΦΜ θα καταχωρεί στη καρτέλα του ασθενή τις εξετάσεις που έχει κάνει μαζί με τη χρονολογία όπου έγιναν.

## 4.4 Αναζήτηση εξετάσεων σε καρτέλα ασθενούς

### Περιγραφή και προτεραιότητα

Σε περίπτωση που θέλει ο χρήστης να κάνει αναζήτηση σε μια καρτέλα κάποιου ασθενή ώστε να βρει τις εξετάσεις που θέλει θα χρειαστεί να πατήσει ένα κλικ στο κουμπί **Search** όπου αυτόματα θα δημιουργηθεί ένα νέο παράθυρο στο οποίο μπορεί να κάνει αναζήτηση των εξετάσεων με τα εξής πεδία:

- *ΑΦΜ*
- *Χρονολογία*

Προτεραιότητα: Υψηλή– 1

### Ακολουθίες ερεθίσματος / απόκρισης

**Ερέθισμα:** Ο χρήστης θα πατήσει το κουμπί **Search** για να κάνει αναζήτηση εργαστηριακών εξετάσεων.

**Απόκριση:** Το πρόγραμμα θα ανοίξει αυτόματα ένα νέο παράθυρο όπου θα ζητά ο χρήστης να δώσει την χρονολογία που έγιναν οι εξετάσεις και το ΑΦΜ του χρηστή ώστε να βρει τον ιατρικό του φάκελο.

**Ερέθισμα:** Ο χρήστης θα εισάγει τα στοιχεία που χρειάζεται και έπειτα θα πατήσει το κουμπί **Αναζήτηση**.

**Απόκριση:** Το σύστημα με βάση τον ΑΦΜ και τη χρονολογία θα κάνει την αναζήτηση του χρηστή στη βάση δεδομένων του προγράμματος και αν βρει θα εμφανίσει τις εξετάσεις με βάση τη χρονολογία που έδωσε ο χρήστης αλλιώς θα εμφανίσει όλες τις εξετάσεις και όχι μόνο της συγκεκριμένης χρονολογίας. Επίσης εμφανίζονται και τα πλήρη στοιχεία του. Σε διαφορετική περίπτωση εμφανίζεται μήνυμα αποτυχίας.

## 4.5 Αλλαγή (διόρθωση) στοιχείων ασθενούς

### Περιγραφή και προτεραιότητα

Σε περίπτωση που θέλει ο χρήστης να κάνει κάποια αλλαγή (διόρθωση) στα στοιχεία ενός ήδη υπάρχοντος χρηστή θα χρειαστεί να πατήσει ένα κλικ στο κουμπί **Edit** αυτόματα θα δημιουργηθεί ένα νέο παράθυρο στο οποίο μπορεί να κάνει αλλαγή των στοιχείων ενός ασθενούς με τα εξής πεδία:

- Επιλογή πεδίου προς αλλαγή
- Αναζήτηση ασθενούς μέσω ΑΦΜ
- Εισαγωγή δεδομένου προς ανανέωση

Προτεραιότητα: Μέτρια – 2.

### Ακολουθίες ερεθίσματος / απόκρισης

**Ερεθίσμα:** Ο χρήστης θα πατήσει το κουμπί **Edit** για να κάνει αλλαγή (διόρθωση) στα στοιχεία ενός ασθενή.

**Απόκριση:** Το πρόγραμμα θα ανοίξει αυτόματα ένα νέο παράθυρο όπου θα ζητά ο χρήστης να δώσει:

- I) Το ΑΦΜ όπου θα με αυτό τον τρόπο θα κάνει την αντιστοιχία με τον ασθενή που θέλουμε .
- II) Πιο πεδίο θέλουμε να αλλάξουμε στον ασθενή  
**Π.χ.** Σε περίπτωση που θέλει ο χρήστης να αλλάξει το όνομα θα επιλέξει name.
- III) Το νέο δεδομένο που θέλουμε να εισάγουμε  
**Π.χ.** Σε περίπτωση που θέλει ο χρήστης να αλλάξει το όνομα του ασθενή γιατί είχε γίνει κάποιο λάθος στην εγγραφή του δεν έχει πάρα να γράψει το σωστό όνομα και να πατήσει το κουμπί αλλαγής.

## 5. Μη λειτουργικές απαιτήσεις

### 5.1 Επιδόσεις

- Η συναλλαγή των δεδομένων μεταξύ server – client θα γίνεται σε λιγότερο από 5 δευτερόλεπτα για κάθε χρήστη.
- Ο μέγιστος αριθμός χρηστών ανέρχεται στα 1000 ασύγχρονα συνδεδεμένα άτομα

### 5.2 Ασφάλεια πληροφορίας

Οι χρηστές θα έχουν τον δικό τους κωδικό πρόσβασης ο οποίος διαφέρει για κάθε κατηγορία χρήστη. Τα δεδομένα που υπάρχουν στη βάση θα γίνονται back-up κάθε μήνα έτσι ώστε να μην χάνονται οι εγγραφές. Όσο αναφορά την ασφάλεια απο εξωτερικούς παράγοντες το πρόγραμμα διαθέτει :

- Πρωτόκολλο SSL
- Τεχνολογία VPN

### 5.3 Ποιότητα λογισμικού

- Δυνατότητα συντήρησης

*Το λογισμικό έχει τη δυνατότητα συντήρησης ώστε σε περίπτωση σφάλματος σε οποιαδήποτε κομμάτι της εφαρμογής να είναι δυνατή η άμεση αποκατάστασή του.*

- Δυνατότητα αναβάθμισης

*Το λογισμικό έχει τη δυνατότητα αναβάθμισης ώστε στο μέλλον να μπορούν να προστεθούν και άλλες λειτουργίες για τις ανάγκες του νοσοκομείου.*

- Αξιοπιστία

*Το σύστημα θα είναι διαθέσιμο για το χρηστή οποιαδήποτε στιγμή το χρειαστεί και η συχνότητα αποτυχίας στη χρήση του είναι μηδενική εκτός αν έχει γίνει λάθος εισαγωγή από τον ίδιο χρηστή.*

- Ευχρηστία

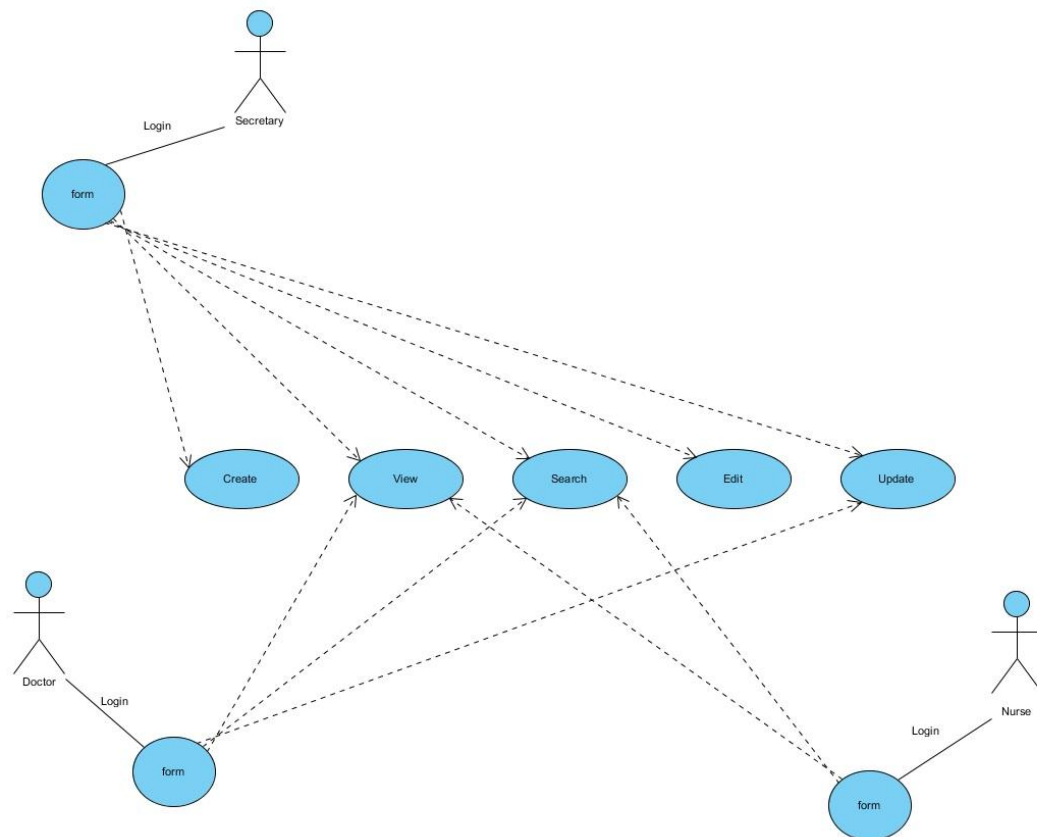
*Το λογισμικό διαθέτει ένα ιδιαίτερα φιλικό περιβάλλον προς το χρηστή και δε χρειάζεται να είναι εξειδικευμένος με τους Η/Υ. Για την εξοικείωση του χρήστη με το λογισμικό χρειάζεται ελάχιστη ώρα.*

- Ικανότητα επαναχρησιμοποίησης

*Η εγκατάσταση του προγράμματος δεν απαιτεί περαιτέρω εγκαταστάσεις προγραμμάτων, οπότε σε περίπτωση που χρειαστεί να εγκατασταθεί η να επανεγκατασταθεί απο την αρχή χρειάζεται μόνο το εκτελέσιμο αρχείο.*

## 6. ΔΙΑΓΡΑΜΜΑ ΠΕΡΙΠΤΩΣΗΣ ΧΡΗΣΤΗ

Κάθε χρήστης έχει άμεση συσχέτιση με το log-in αλλά έχει διαφορετική εξάρτηση από τις λειτουργίες που υπάρχουν. Τα βελάκια με τις διακεκομμένες γραμμές υποδεικνύουν την εξάρτηση ενώ τα συνεχόμενα την συσχέτιση.

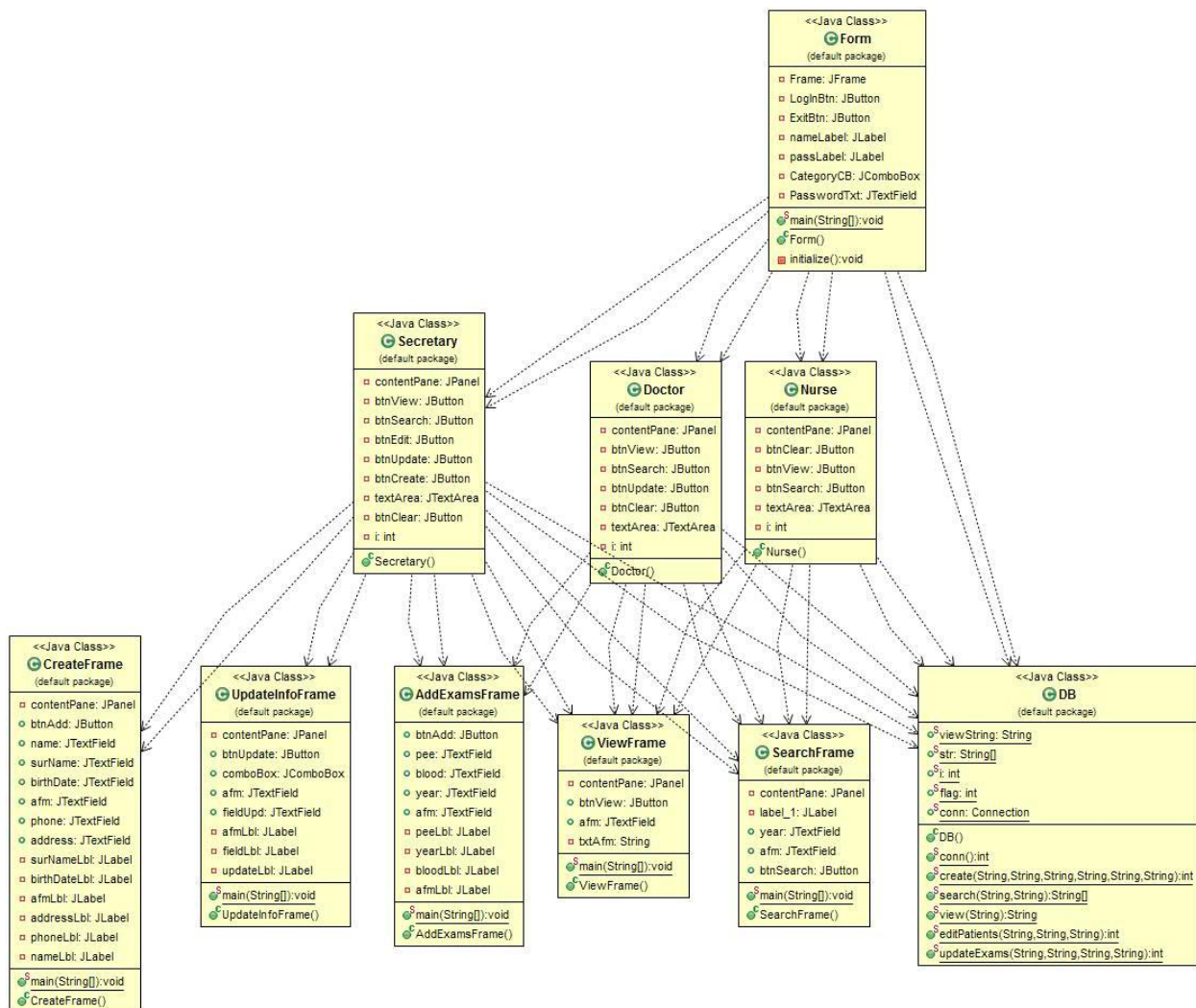


## 7. Διαγράμματα UML

Στην εργασία συμπεριλαμβάνονται 1 διάγραμμα κλάσεων , 3 διαγράμματα δραστηριότητας και 3 διαγράμματα ακολουθίας.

### 7.1 Διάγραμμα κλάσεων (class diagram)

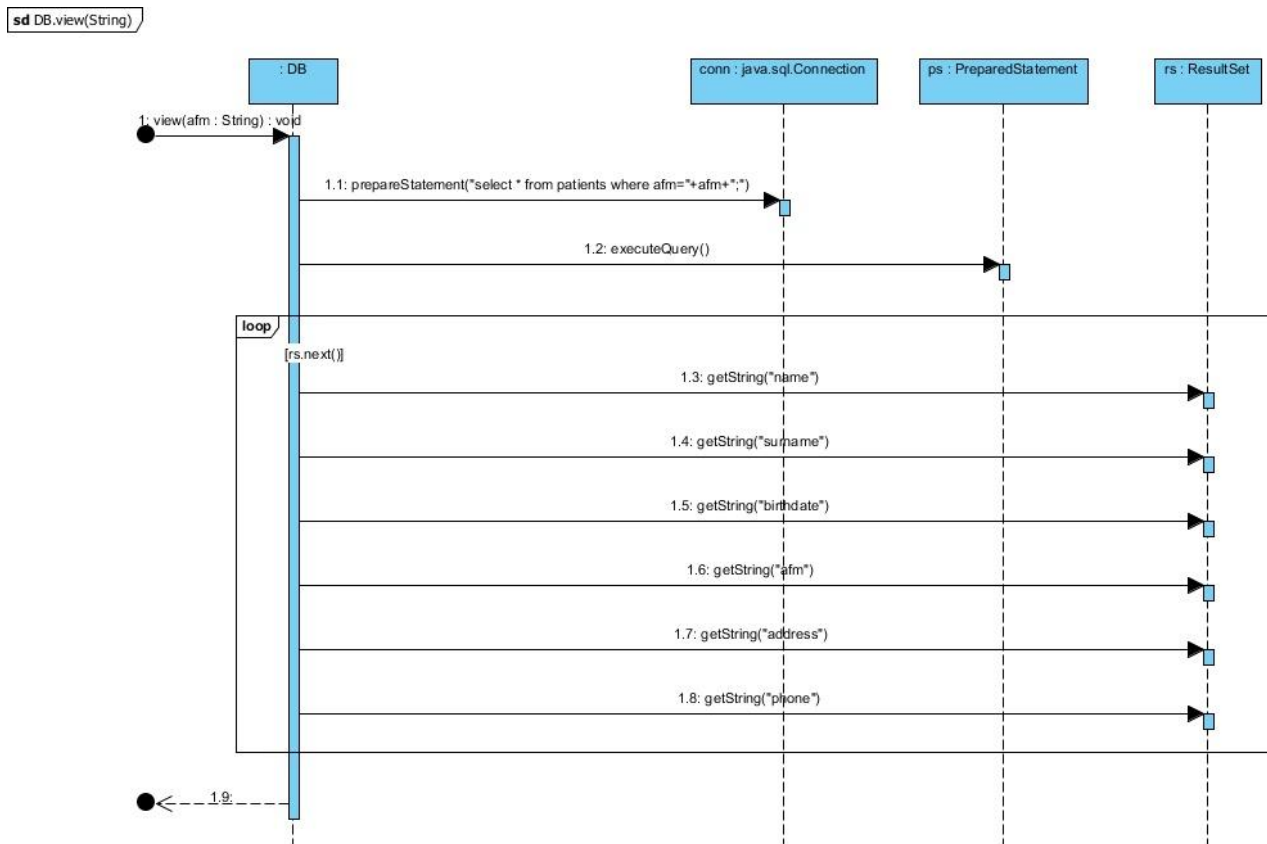
Το διάγραμμα κλάσεων έγινε auto-generate από plug-in που διαθέτει το eclipse.



## 7.2 Διαγράμματα ακολουθίας (sequence diagram)

Τα διαγράμματα ακολουθίας έγιναν με την βοήθεια του προγράμματος Visual Paradigm.

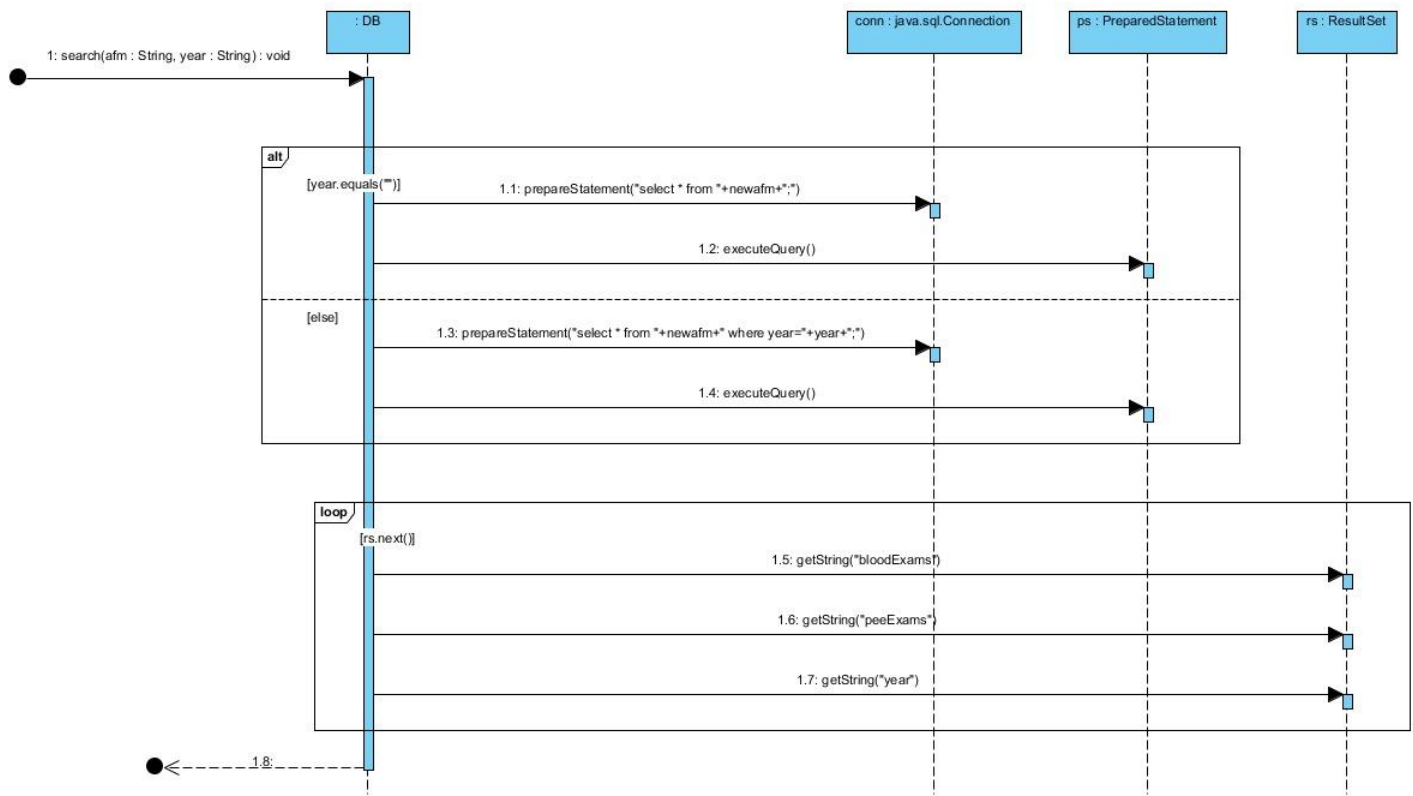
Το πρώτο διάγραμμα υλοποιεί την λειτουργία view:





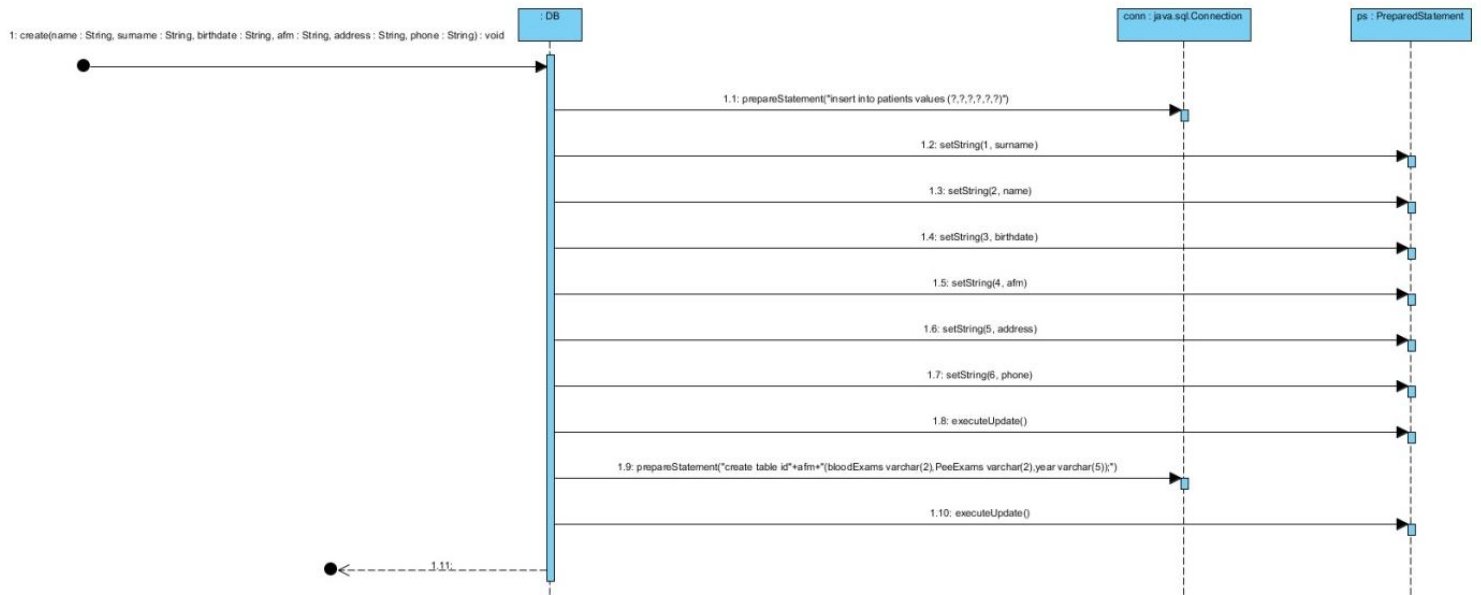
Το δεύτερο διάγραμμα υλοποιεί την λειτουργία search:

sd DB.search(String, String)



Το τρίτο διάγραμμα υλοποιεί την λειτουργία create:

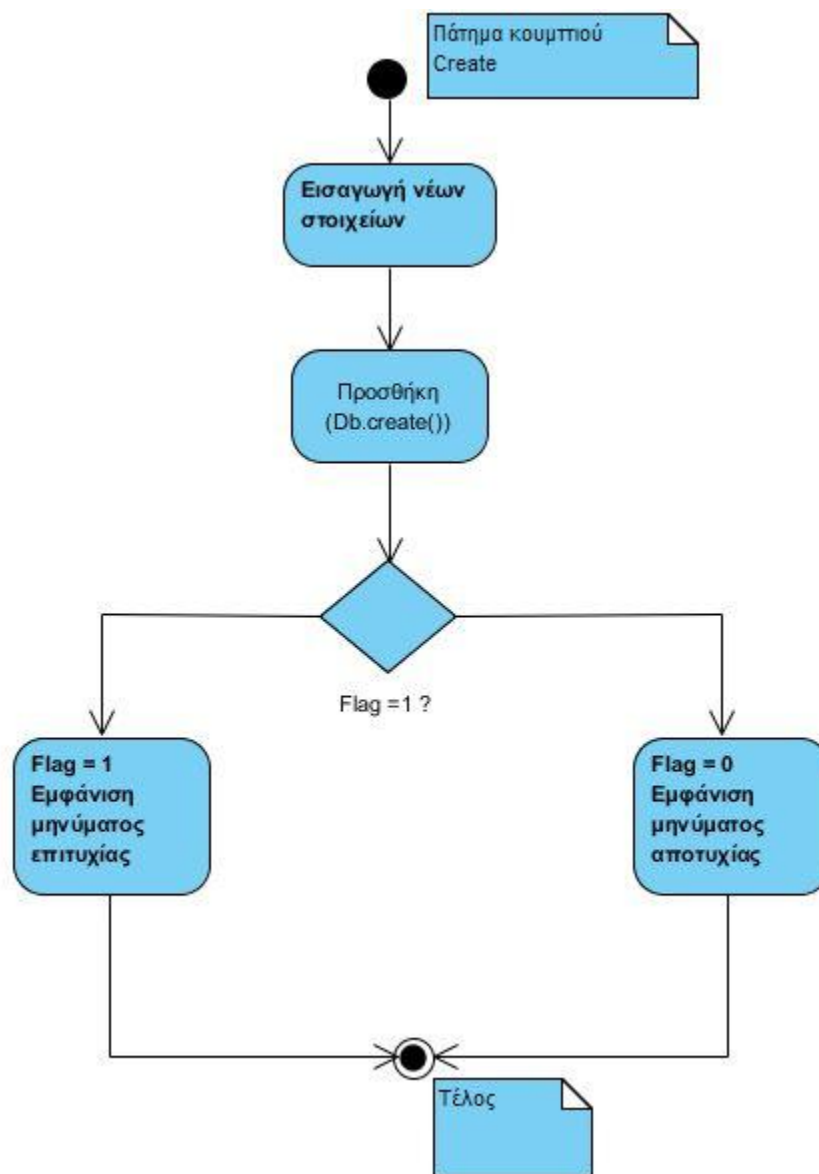
sd DB.conn()



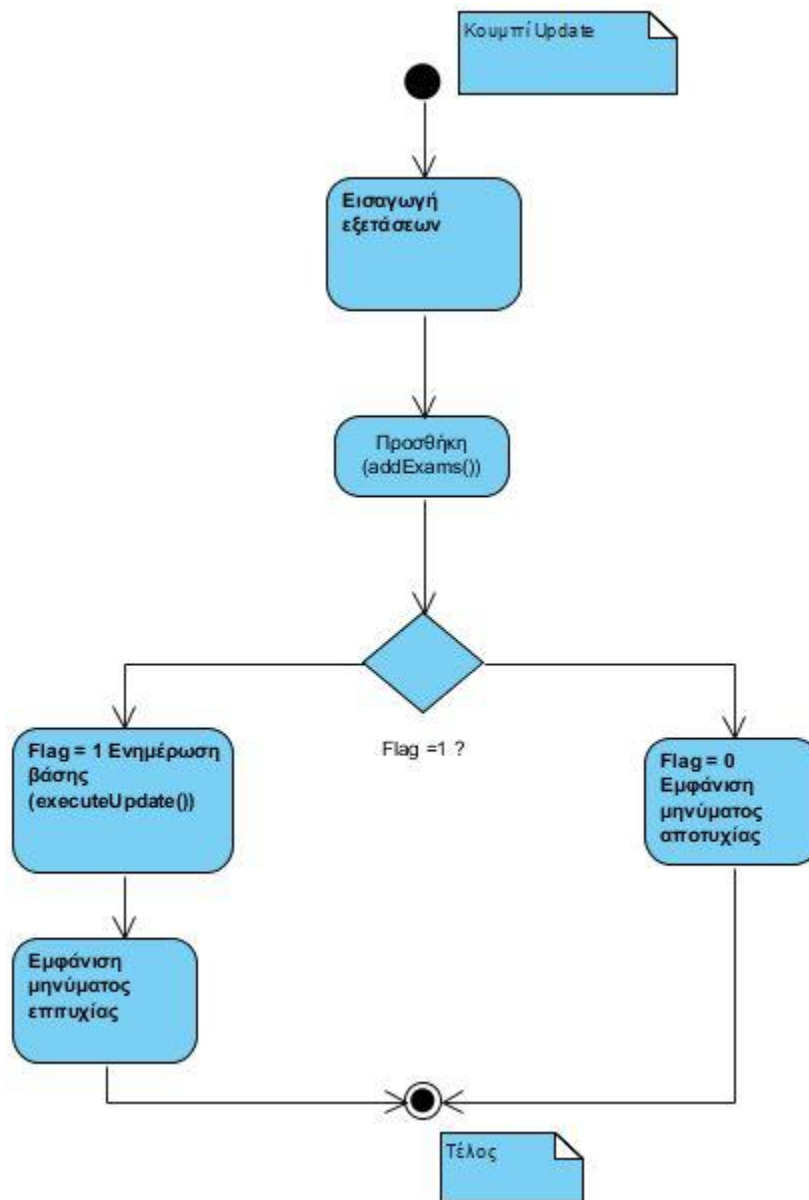
### 7.3 Διαγράμματα δραστηριότητας

Και τα διαγράμματα δραστηριότητας υλοποιήθηκαν με την βοήθεια του προγράμματος Visual Paradigm.

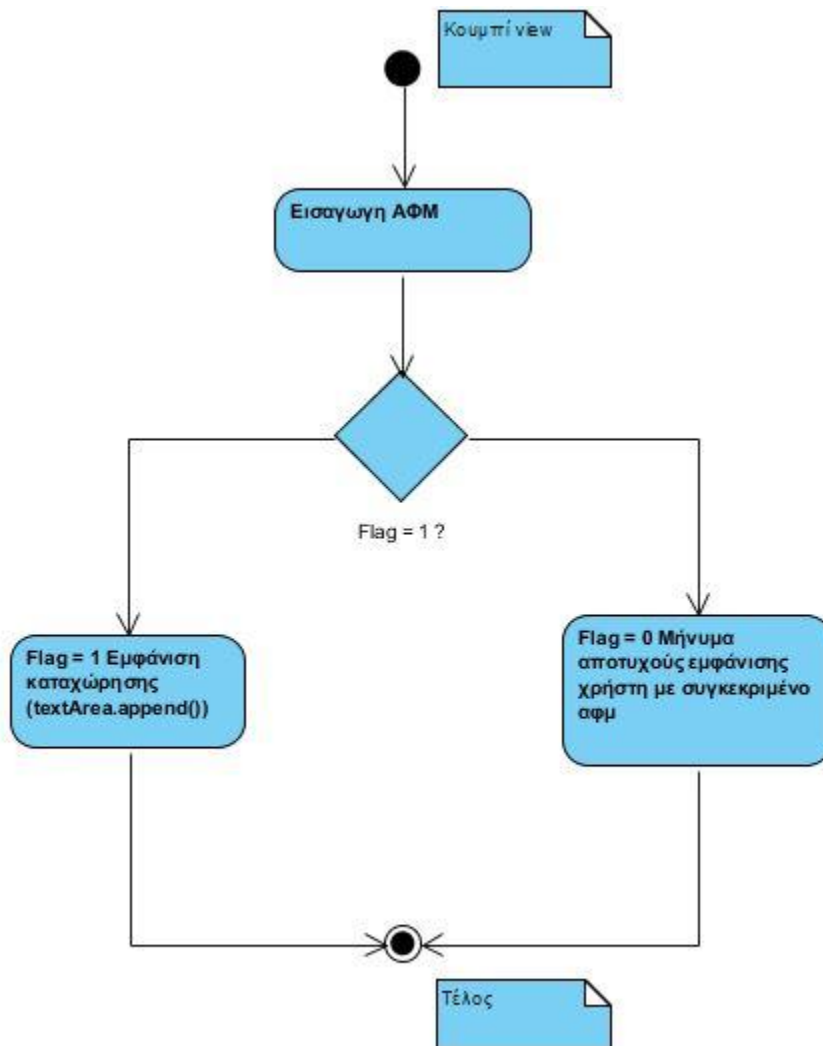
Το πρώτο διάγραμμα υλοποιεί την λειτουργία create:



Το δεύτερο διάγραμμα υλοποιεί την λειτουργία update:



Το τρίτο διάγραμμα υλοποιεί την λειτουργία view:



## 8. PMD TESTING

Μετά το PMD έλεγχο αφαιρέθηκαν τα κόκκινα violations τα οποία φαίνονται στις εικόνες παρακάτω

Package Explorer:

- ML
  - src
    - (default package)
      - AddExamsFrame.java
      - CreateFrame.java
      - DB.java
      - Doctor.java
      - Form.java
      - Nurse.java
      - SearchFrame.java
      - Secretary.java
      - UpdateInfoFrame.java
      - ViewFrame.java
    - JRE System Library [JavaSE-1.7]
    - Referenced Libraries

Violations Outline:

Line	Rule	Error Message
86	VariableNamingCo...	Variables should start with a lowercase chara...
23	VariableNamingCo...	Variables should start with a lowercase chara...
26	VariableNamingCo...	Variables should start with a lowercase chara...
20	VariableNamingCo...	Variables should start with a lowercase chara...
27	VariableNamingCo...	Variables should start with a lowercase chara...
22	VariableNamingCo...	Variables should start with a lowercase chara...
52	VariableNamingCo...	Variables should start with a lowercase chara...
85	VariableNamingCo...	Variables should start with a lowercase chara...
23	CommentRequired	fieldCommentRequirement Required
27	BeanMembersShoul...	Found non-transient, non-static member. Pl...
27	CommentRequired	fieldCommentRequirement Required
53	LawOfDemeter	Potential violation of Law of Demeter (meth...
58	LocalVariableCould...	Local variable 'categ' could be declared final

Violations Overview:

Element	# Violations	# Violations/K...	# Violations/M...	Project
(default package)	591	731.4	26.86	ML
CreateFrame.java	56	746.7	28.00	ML
UpdateInfoFrame.java	36	562.5	18.00	ML
DB.java	120	1016.9	20.00	ML
ViewFrame.java	25	833.3	12.50	ML
SearchFrame.java	30	769.2	15.00	ML
Secretary.java	100	662.3	100.00	ML
Doctor.java	70	642.2	70.00	ML
AddExamsFrame.java	39	722.2	19.50	ML

Package Explorer:

- ML
  - src
    - (default package)
      - AddExamsFrame.java
      - CreateFrame.java
      - DB.java
      - Doctor.java
      - Form.java
      - Nurse.java
      - SearchFrame.java
      - Secretary.java
      - UpdateInfoFrame.java
      - ViewFrame.java
    - JRE System Library [JavaSE-1.7]
    - Referenced Libraries

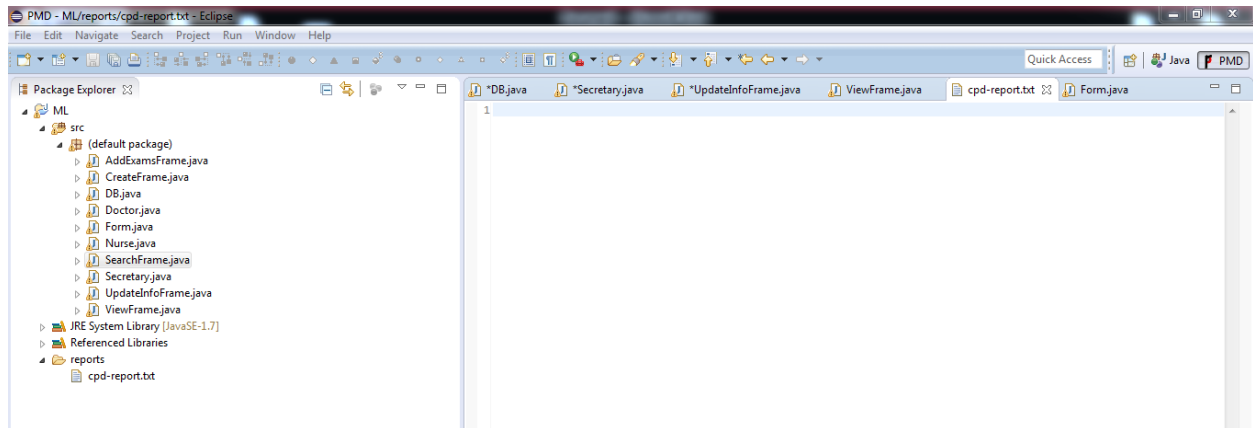
Violations Outline:

Line	Rule	Error Message
21	BeanMembersShoul...	Found non-transient, non-static member. Please mark as...
17	NoPackage	All classes and interfaces must belong to a named package
19	CommentRequired	fieldCommentRequirement Required
87	LocalVariableCouldBeF...	Local variable 'sectr' could be declared final
98	UseEqualsToCompareS...	Use equals() to compare strings instead of '==' or '!='
85	LocalVariableCouldBeF...	Local variable 'categoryPass' could be declared final
24	CommentRequired	fieldCommentRequirement Required
89	LocalVariableCouldBeF...	Local variable 'docstr' could be declared final
41	CommentRequired	publicMethodCommentRequirement Required
84	LocalVariableCouldBeF...	Local variable 'categoryName' could be declared final
24	SingularField	Perhaps 'passLabel' could be replaced by a local variable.
80	CommentRequired	publicMethodCommentRequirement Required

Violations Overview:

Element	# Violations	# Violations/KLOC	# Violations/Method	Project
(default package)	577	714.1	26.23	ML
AddExamsFrame.java	39	722.2	19.50	ML
CreateFrame.java	56	746.7	28.00	ML
DB.java	119	1008.5	19.83	ML
Doctor.java	70	642.2	70.00	ML
Form.java	54	658.5	18.00	ML
Nurse.java	52	604.7	52.00	ML

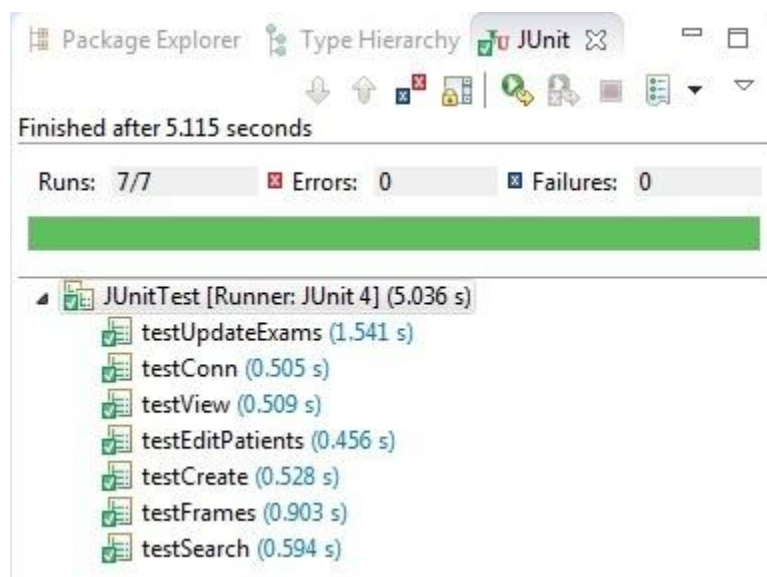
Έγινε επίσης έλεγχος για copy & paste μέσα στον κώδικα χωρίς όμως αποτελέσματα που σημαίνει ότι δεν υπήρξαν αντιγραφές κομματιών του κώδικα.



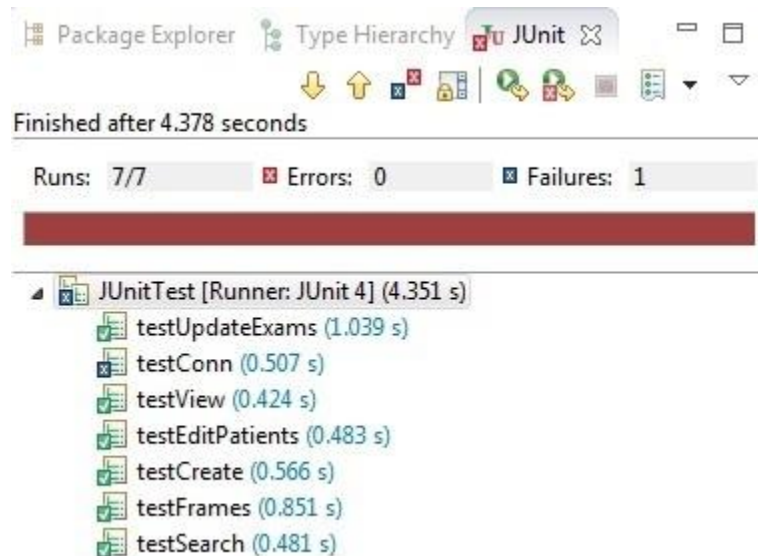
## 9. UNIT TESTING & CODE COVERAGE

Παρακάτω ακολουθούν τα αποτελέσματα του unit testing και του code coverage. Για την πλήρη κατανόηση της μεθοδολογίας πήραμε 2 περιπτώσεις. Στην πρώτη δεν υπάρχει κανένα λάθος, που είναι ο κανονικός κώδικας χωρίς λάθη, ενώ στη δεύτερη προστέθηκε επίτηδες λάθος.

Επιτυχημένο unit testing:



Αποτυχημένο unit testing:



Έπειτα ακολουθούν εικόνες απο τα στατιστικά του code coverage και το highlighted text. Στο highlighted text το πράσινο υποδηλώνει τις γραμμές που εκτελούνται, το κίτρινο τις συνθήκες ενώ το κόκκινο ο κώδικας που δεν συμπεριλαμβάνεται. Στην εικόνα με τα στατιστικά υπάρχει 33.7 % επικάλυψη κώδικα σε όλο το project.

```
9 public class DB {
10
11     public static String viewString ; //Μεταβλητή για την ανάθεση αποτελεσμάτων μετά από queries
12     public static String [] str = new String[20]; //Πίνακας με δεδομένα μετά απο queries που επιστρέφουν πολλαπλές καταχωρήσεις
13     public static int i = 0; //Counter
14     public static int flag; //Μεταβλητή για τον προσδιορισμό επιτυχίας (ή μη) των λειτουργιών
15                                     //Σε περίπτωση επιτυχής λειτουργίας ανατίθεται η τιμή 1.Σε αντίθετη περίπτωση flag
16     public static Connection conn = null;
17
18     public static int conn() {
19         //Σύνδεση με τη βάση δεδομένων
20
21         flag=1;
22         try {
23             Class.forName("com.mysql.jdbc.Driver").newInstance();
24
25             String connection = "jdbc:mysql://localhost:3306/kliniki";
26             String user = "root";
27             String pass = "root";
28
29             conn = DriverManager.getConnection(connection, user, pass);
30
31         } catch (Exception e) {
32             flag=0;
33             e.printStackTrace();
34         }
35         return flag;
36     }
37     public static int create(String name, String surname, String birthdate,String afm, String address, String phone){
38         //Συνάρτηση η οποία δημιουργεί μια καταχώρηση στον πίνακα των ασθενών
39
40         flag =1;
41         PreparedStatement ps = null;
42
43         try {
44             //Δημιουργία του αντίστοιχου query για την προσθήκη των στοιχείων του ασθενούς στον αντίστοιχο πίνακα
45
46             ps = conn.prepareStatement("insert into patients values (?, ?, ?, ?, ?)");
47             ps.setString(1, surname);
48             ps.setString(2, name);
49             ps.setString(3, birthdate);
```



Element	Coverage	Covered Instruction...	Missed Instructions	Total Instructions
JUnitTest (1) (Jan 25, 2015 6:59:46 PM)				
MedicalFileManagement	33.7 %	1,064	2,091	3,155
src	30.6 %	919	2,088	3,007
(default package)	30.6 %	919	2,088	3,007
Secretary.java	42.4 %	256	348	604
CreateFrame.java	0.0 %	0	317	317
Form.java	0.0 %	0	296	296
Doctor.java	46.3 %	192	223	415
UpdateInfoFrame.java	0.0 %	0	222	222
AddExamsFrame.java	0.0 %	0	216	216
Nurse.java	49.5 %	160	163	323
SearchFrame.java	0.0 %	0	150	150
ViewFrame.java	0.0 %	0	109	109
DB.java	87.6 %	311	44	355
unit	98.0 %	145	3	148
(default package)	98.0 %	145	3	148
JUnitTest.java	98.0 %	145	3	148

## 10. DEBUGGING

Παρακάτω ακολουθούν εικόνες για τη λειτουργία του debugging. Στην πρώτη εικόνα φαίνεται το break point πριν γίνει step into στη ρουτίνα εργασίας. Στη δεύτερη εικόνα φαίνεται το αποτέλεσμα του debugging το οποίο επιστρέφει μία καταχώρηση, αφού εκτελέσει την λειτουργία view, η οποία βρίσκεται στην κατηγορία variables πάνω δεξιά.

The screenshot displays the IDE's debugging interface. The top panel shows the 'Debug' view with a breakpoint set at line 22 in DB.java. The middle panel shows the 'Variables' view with the 'args' variable set to 'String[0] (id=20)'. The bottom panel shows the 'DB.java' source code with the breakpoint at line 22. The right panel shows the 'Outline' view with the 'DB' class and its methods.

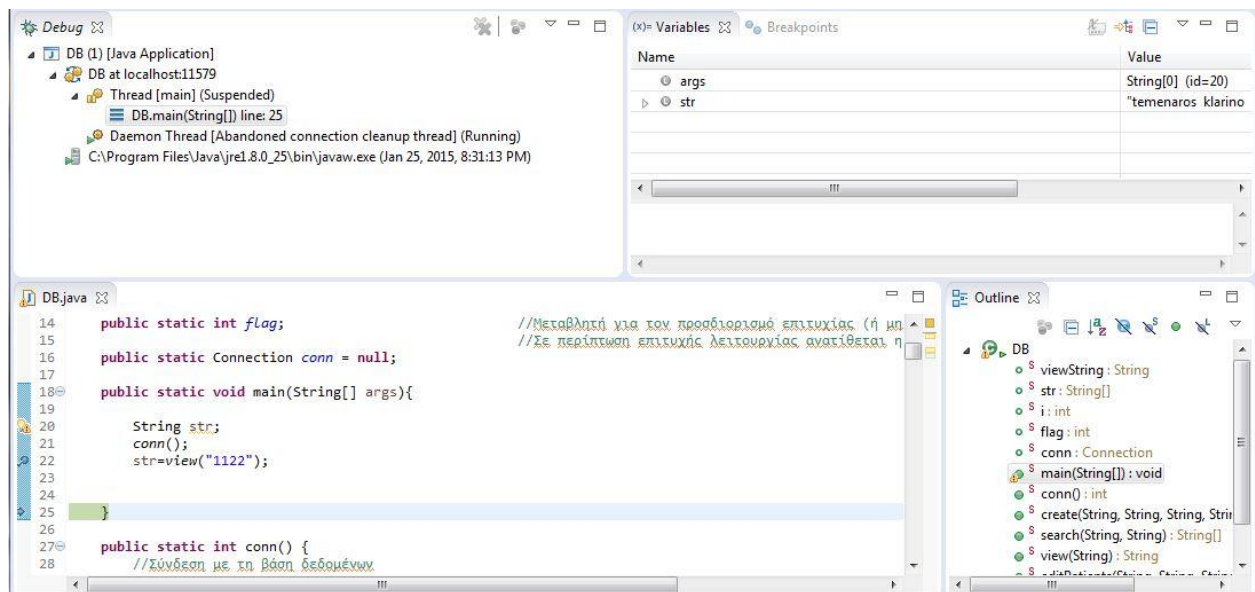
```

11 public static String viewString;
12 public static String [] str = new String[20];
13 public static int i = 0;
14 public static int flag;
15
16 public static Connection conn = null;
17
18 public static void main(String[] args){
19
20     String str;
21     conn();
22     str=view("1122");
23
24
25 }
  
```

Outline:

- DB
  - viewString : String
  - str : String[]
  - i : int
  - flag : int
  - conn : Connection
  - main(String[]): void
  - conn(): int
  - create(String, String, String, String): void
  - search(String, String): String[]
  - view(String): String





Παρατήρηση: Στο κυρίως πρόγραμμα που χρησιμοποιείται δεν γίνεται χρήση της main στη DB.class και ο λόγος που προστέθηκε ήταν αποκλειστικά και μόνο για την λειτουργία του debug.

## 11. User Acceptance Test (UAT) Plan

### 11.1 Document Management

#### 11.1.1 Contributors

*Please provide details of all contributors to this document.*

Role	Department	Name
Data Quality Manager	IT	Λατσίνογλου Νίκος
Senior Software Engineer	IT	Ντόκος Στέφανος
Systems Administrator	IT	Τέμενος Νίκος

### 11.1.2 Version Control

*Please document all changes made to this document since initial distribution.*

Date	Version	Author	Section	Amendment
15/12/2014	1.0	Λατσίνογλου Νίκος	1	Καθορισμός εργασιών για την ομάδα
17/12/2014	1.1	Λατσίνογλου Νίκος	1	Ομαδική σχεδίαση του πλάνου για το project
23/12/2014	2.0	Ντόκος Στέφανος	2	Δημιουργία Βάσης Δεδομένων και σύνδεση με το πρόγραμμα
3/1/2015	2.1	Τέμενος Νίκος	3	Έλεγχος τελικού project για προβλήματα
9/1/2015	2.2	Ντόκος Στέφανος	2	Επιβεβαίωση τελικού project

## 11.2 USER ACCEPTANCE TESTING

### 11.2.1 User Acceptance Definition

User Acceptance Testing should ensure that the application performs at an acceptable level for the Customer.

### 11.2.2 UAT Responsibilities

Role	Name	Responsibilities
Data Quality Manager	Λατσίνογλου Νίκος	Ελέγχει τα backups του server, το up-time του server και βεβαιώνεται ότι δεν θα υπάρξει πρόβλημα μεταξύ προγράμματος – βάσης δεδομένων
Senior Software Engineer	Ντόκος Στέφανος	Δημιουργεί το πρόγραμμα και ελέγχει για bugs στον κώδικα
Systems Administrator	Τέμενος Νίκος	Ελέγχει κατά πόσο εύκολα γίνεται το σύστημα διαπεραστικό από εξωτερικούς παράγοντες. Ελέγχει προβλήματα στον κώδικα

### 11.3 UAT STRATEGY

Consideration	Date Agreed	Details
Test Approach	12/1/2015	Συνάντηση της ομάδας για επιβεβαίωση από όλους για το project
Assumptions & Constraints	13/1/2015	Οι δοκιμές του προγράμματος θα διαρκέσουν μέχρι να είναι σίγουρη η ασφαλής λειτουργία του προγράμματος
Setup of Test Environment	15/1/2015	Εγκατάσταση του προγράμματος στους υπολογιστές της κλινικής και σύνδεση με το server
Test Scenarios	15/1/2015	Πρώτες δοκιμές εισαγωγής χρηστών με εξετάσεις και εμφάνιση των στοιχείων τους
Test Period	16/1/2015 – 21/1/2015	Έναρξη και λήξη δοκιμών του προγράμματος

### 11.4 USER TESTING

The following explicit tests cover the main areas of work as load out in section 3 above.

1.	Test Case			
Test #	Tasks	Expected Result	Notes	Pass /Fail
1.1	Είσοδος με σωστά στοιχεία σε κάθε κατηγορία χρηστών	Ανακατεύθυνση σε διαφορετική φόρμα για κάθε χρήστη	Μετά την εισαγωγή των κωδικών, ο χρήστης οδηγείται στην δική του φόρμα με συγκεκριμένες λειτουργίες	PASS
1.2	Είσοδος με στοιχεία διαφορετικής ομάδας χρηστών	Μη δυνατή πρόσβαση στο σύστημα	Επιβεβαίωση ότι μια κατηγορία χρηστών δεν θα έχει πρόσβαση σε άλλες	FAIL

			λειτουργίες του προγράμματος	
<b>1.3</b>	Είσοδος με τυχαία στοιχεία χρήστη	Μη δυνατή πρόσβαση στο σύστημα	Επιβεβαίωση ότι ο χρήστης θα εισάγεται στο σύστημα μόνο με τα δικά του στοιχεία	FAIL

<b>2.</b>	<b>Test Case</b>			
<b>Test #</b>	<b>Tasks</b>	<b>Expected Result</b>	<b>Notes</b>	<b>Pass /Fail</b>
<b>2.1</b>	Καταχώρηση στοιχείων ασθενούς από τη γραμματεία	Εμφάνιση μηνύματος επιτυχίας	Αν η εισαγωγή χρήστη γίνει με μοναδικό ΑΦΜ τότε εμφανίζεται μήνυμα επιτυχίας	PASS
<b>2.2</b>	Εμφάνιση στοιχείων ασθενούς απο γραμματεία	Εμφάνιση στοιχείων	Με βάση το ΑΦΜ δίνονται τα στοιχεία του ασθενούς	PASS
<b>2.3</b>	Εμφάνιση εξετάσεων απο γραμματεία	Εμφάνιση χρονολογίας εξετάσεων ασθενούς	Με βάση το ΑΦΜ και χρονολογία δίνονται τα στοιχεία των εξετάσεων του ασθενή	PASS